

Document made available under the Patent Cooperation Treaty (PCT)

International application number: PCT/JP05/001833

International filing date: 08 February 2005 (08.02.2005)

Document type: Certified copy of priority document

Document details: Country/Office: JP
Number: 2004-031761
Filing date: 09 February 2004 (09.02.2004)

Date of receipt at the International Bureau: 07 April 2005 (07.04.2005)

Remark: Priority document submitted or transmitted to the International Bureau in compliance with Rule 17.1(a) or (b)



World Intellectual Property Organization (WIPO) - Geneva, Switzerland
Organisation Mondiale de la Propriété Intellectuelle (OMPI) - Genève, Suisse

10.02.2005

日 本 国 特 許 庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日
Date of Application: 2 0 0 4 年 2 月 9 日

出 願 番 号
Application Number: 特 願 2 0 0 4 - 0 3 1 7 6 1
[ST. 10/C]: [J P 2 0 0 4 - 0 3 1 7 6 1]

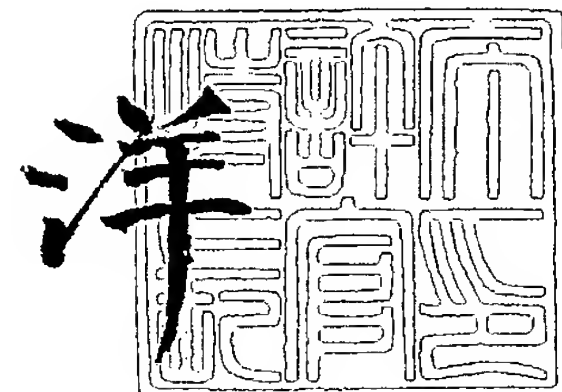
出 願 人
Applicant(s): 松下電器産業株式会社



2 0 0 5 年 3 月 2 4 日

特許庁長官
Commissioner,
Japan Patent Office

小 川



【書類名】 特許願
【整理番号】 2048260008
【提出日】 平成16年 2月 9日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 15/00
【発明者】
 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
 【氏名】 金丸 智一
【発明者】
 【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内
 【氏名】 大蘆 雅弘
【特許出願人】
 【識別番号】 000005821
 【氏名又は名称】 松下電器産業株式会社
【代理人】
 【識別番号】 100097445
 【弁理士】
 【氏名又は名称】 岩橋 文雄
【選任した代理人】
 【識別番号】 100103355
 【弁理士】
 【氏名又は名称】 坂口 智康
【選任した代理人】
 【識別番号】 100109667
 【弁理士】
 【氏名又は名称】 内藤 浩樹
【手数料の表示】
 【予納台帳番号】 011305
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 9809938

【書類名】 特許請求の範囲**【請求項 1】**

アプリケーションを実行する、アプリ実行手段と、

別の機器上に存在するアプリケーションの特徴を認識する、アプリ認識手段と、

別の機器上に存在するアプリケーションが実行不能になったことを検出する、アプリ使用不能検出手段と、

別の機器上に存在するアプリケーションと同一のアプリケーションが、機器に存在するかどうかを判定する、アプリ同一性判定手段と

別の機器上に存在するアプリケーションに対する起動要求を検出する、アプリ起動要求検出手段とを備え、

別のアプリ実行手段を備える 1 つ以上の別の機器と、ネットワークを介して接続されており、

アプリ使用不能検知手段が検知した、別の機器上で実行不能になったアプリケーションと同一のアプリケーションを機器が保持しているかどうかをアプリ同一性判定手段によって判定し、

保持している場合には、アプリ起動要求検出手段により検出した別の機器に対するアプリケーションの起動要求をトリガとして、自機器が同一のアプリケーションの実行を開始することを特徴とした、ネットワークに接続される機器。

【請求項 2】

アプリケーションを実行する、アプリ実行手段と、

別の機器上に存在するアプリケーションの特徴を認識する、アプリ認識手段と、

別の機器上に存在するアプリケーションが実行不能になったことを検出する、アプリ使用不能検出手段と、

別の機器上に存在するアプリケーションと同一のアプリケーションが、機器に存在するかどうかを判定する、アプリ同一性判定手段と

別の機器上で実行されているアプリケーションの状態を取得する、アプリ実行状態取得手段と、を備え、

別のアプリ実行手段を備える 1 つ以上の別の機器と、ネットワークを介して接続されており、

アプリ使用不能検知手段が検知した、別の機器上で実行不能になったアプリケーションと同一のアプリケーションを機器が保持しているかどうかをアプリ同一性判定手段によって判定し、

保持している場合には、別の機器のアプリ実行状態通知手段によって通知されたサービス状態を、アプリ実行状態取得手段により取得し

取得したサービス状態を利用して、自機器が同一のアプリケーションを自動的に実行することを特徴とした、ネットワークに接続される機器。

【請求項 3】

アプリケーションを実行する、アプリ実行手段と、

別の機器上に存在するアプリケーションの特徴を認識する、アプリ認識手段と、

別の機器上に存在するアプリケーションが実行不能になったことを検出する、アプリ使用不能検出手段と、

別の機器上に存在するアプリケーションに対する起動要求を検出する、アプリ起動要求検出手段と、

別の機器上に存在するアプリケーションと関係付けされた、機器が保持する別のアプリケーションを検索し特定する、代替アプリ判定手段と、を備え、

別のアプリ実行手段を備える 1 つ以上の別の機器と、ネットワークを介して接続されており、

アプリ使用不能検出手段が検知した、別の機器上で実行不能になったアプリケーションと関係付けされたアプリケーションを、機器が保持しているかどうかを代替アプリ判定手段によって判定し、

保持している場合には、アプリ起動要求検出手段により検出した別の機器に対するアプリケーションの起動要求をトリガとして、自機器が代替アプリケーションを自動的に実行することを特徴とした、ネットワークに接続される機器。

【請求項 4】

アプリケーションを実行する、アプリ実行手段と、

別の機器上に存在するアプリケーションの特徴を認識する、アプリ認識手段と、

別の機器上に存在するアプリケーションが実行不能になったことを検出する、アプリ使用不能検出手段と、

別の機器上で実行されているアプリケーションの状態を取得する、アプリ実行状態取得手段と、

別の機器上に存在するアプリケーションと関係付けされた、機器が保持する別のアプリケーションを検索し特定する、代替アプリ判定手段と、を備え、

別のアプリ実行手段を備える 1 つ以上の別の機器と、ネットワークを介して接続されており、

アプリ使用不能検出手段が検知した、別の機器上で実行不能になったアプリケーションと関係付けされたアプリケーションを、機器が保持しているかどうかを代替アプリ判定手段によって判定し、

保持している場合には、アプリ実行状態取得手段により取得したサービス状態を利用して、自機器が代替アプリケーションを自動的に実行する

ことを特徴とした、ネットワークに接続される機器。

【書類名】 明細書

【発明の名称】 機器状態に応じサービスを自動継続する装置

【技術分野】

【0 0 0 1】

本発明はネットワークにより相互接続された家電機器上で、ソフトウェアプログラム（アプリケーション）を動作させることによって、サービスを実施するシステムに関するものである。

【背景技術】

【0 0 0 2】

ネットワークに接続された家電機器上で、アプリケーションを動作させるアプリケーション実行系の技術については、サンマイクロシステムズ社の J A V A (R) や、マイクロソフト社の . N E T (ドットネット) などがある。

【0 0 0 3】

これらの技術は仮想マシンの仕組みを備え、ネットワーク家電の C P U など、ハードウェアに依存せずに同一のアプリケーションを動作させることができることから、サービスの実施のための共通プラットフォームとなることが期待されている。

【0 0 0 4】

またネットワークを通じて、アプリケーションを実行する環境の自動設定を行う技術としては、マイクロソフト社によって提唱されている技術である U P n P (U n i v e r s a l P l u g a n d P l a y) や、サンマイクロシステムズ社によって提唱されている技術である J i n i (TM) などが存在している。

【0 0 0 5】

これらの技術の特徴として、ネットワークに機器を繋ぐだけで、ユーザが複雑な設定作業を行わなくともすぐにネットワーク機能を利用可能な状態にすることができる、という利便性をユーザに対して提供する、という事が挙げられる。

【0 0 0 6】

ユーザの利用する機器からネットワークを通じて、ユーザ要求に対応する機能を持つ機器を探し出すシステムの例としては、特許文献 1 に示すものが提示されている。

【0 0 0 7】

サービスに関して何らかの障害が発生した場合に、サーバ機器によって復旧を実施するシステムの例としては、特許文献 2 に示すものが提示されている。

【0 0 0 8】

また、ユーザの利用したい端末でサービスを実施できなかった場合に、代替となる機器を探し出し実施する方法の例としては、特許文献 3 に示すものが提示されている。

【特許文献 1】 特開 2 0 0 3 - 1 7 8 0 3 6 号公報

【特許文献 2】 特開 2 0 0 2 - 2 4 0 3 9 号公報

【特許文献 3】 特開 2 0 0 2 - 9 4 9 1 2 号公報

【発明の開示】

【発明が解決しようとする課題】

【0 0 0 9】

家電機器は、アプリケーションの実行を行うことができない状態に陥ることが頻繁に起こる。

【0 0 1 0】

代表的なケースとしては、機器の電源が切断される、（モバイル機器等において）バッテリーが消費される、などの事態により通電状態が確保できなくなる場合である。他には、同じ機器上で、優先度の高い他のアプリケーションが実行される場合、主に資源の欠乏の問題から、別のアプリケーションの実行を行うことができない場合がある。

【0 0 1 1】

こういった状態に家電機器が陥ると、新たにサービスを家電機器によって実施しようとしても、アプリケーションの起動ができずにサービスの提供が不可能になる。また、例え

アプリケーションを実行中であっても、その実行を中断してサービスの提供を停止せざるを得なくなる。

【0 0 1 2】

現在、サービスの実施が不可能な状態に陥った機器に対して、家電機器がネットワークに接続されているという特性を利用し、サービス実施のための復旧を試みたり、そのサービスの代替実施手段を提供したりすることを目的にした、幾通りかの技術が考案されている。

【0 0 1 3】

特許文献 1 にて示されている従来の技術は、家電ネットワークシステムにおけるユーザ環境の変化やユーザ要求を考慮に入れたシステムを提案しているものの、上記のようなサービスの実施が不可能な状態に陥った場合の具体的な処理方法に関しては言及していないため、サービスの実施の確実性を高めたい場合には、別途具体的な処理手段を実装する必要がある。

【0 0 1 4】

また、特許文献 2 に示されている従来の技術は、サービスの実施が不可能な状態に陥った場合、センターサーバが障害対策と復旧処理を行うシステムについて述べている。しかしこれはあくまでセンターサーバの存在を前提としたものであり、センターサーバの存在を仮定しない、例えば家電機器のピア・ツー・ピアのネットワーク構成のみを前提としたネットワークには対応することができない。

【0 0 1 5】

また、特許文献 3 にて示されている従来の技術は、利用予約した機器に不具合があり、サービスの実施が不可能な状態に陥った場合に、代替となる機器を探し出し利用予約を実施する方法について述べているが、予め機器の組み合わせによって、代替アクションの登録を行っておく必要があった。また、何らかのサービスの実施の最中に、そのサービスの継続が不可能な状態に陥った場合に、そのサービスを継続するための対応に関しては言及されていなかった。

【0 0 1 6】

上記のように、センターサーバによる集中管理方式を前提としない、機器同士のみにより構成されたネットワーク上でも、可能な限りユーザの手を煩わせることなく、サービスの安定性を確保することを目的とする技術に関しては、とりわけ、何らかのサービスの実施中にそのサービスの継続が不可能な状態に陥った場合の復旧手段に関しては、これまでは有効な手段が考えられてはいなかった。

【課題を解決するための手段】

【0 0 1 7】

本発明の機器は、1 つ以上の別の機器とネットワークを介して相互接続される機器であって、アプリケーションを実行する、アプリ実行手段を備える。この際、別の機器上も同一のアプリ実行手段（サービスプラットフォーム）を備え、同じアプリケーションのサービスが動作することを前提としている。

【0 0 1 8】

本発明の機器は、別の機器上に存在するアプリケーションの保持する特徴を認識する、アプリ認識手段と、別の機器上に存在するアプリケーションが、別の機器上で実行不能になったことを検出する、アプリ使用不能検出手段を備える。これにより、本発明の機器は、別の機器が保持しているサービスと、その機器がサービスを実施可能な状態下にあるかどうかを判定することができる。

【0 0 1 9】

また本発明の機器は、別の機器上で実行されているアプリケーションの状態を取得する、アプリ実行状態取得手段を備える。これにより、本発明の機器は、別な機器で実行中であるサービスの状態を取得することができる。

【0 0 2 0】

さらに本発明の機器は、別の機器上に存在するアプリケーションと同一のアプリケーシ

ョンを機器が保持しているかどうかを判定する、アプリ同一性判定手段を備える。

【0021】

別の機器上で実行不能になったアプリケーションと、同一と判断されるアプリケーションを機器が保持している場合には、別の機器に対して何らかの形でアプリケーションの実行要求が入ったにも関わらず、別の機器がそのアプリケーションを実行できない状態にある場合、別の機器に代わって自機器が保持する同一のアプリケーションを自動的に実行することを特徴とする。これを実現するために、本発明の機器はアプリ起動要求検出手段を備える。

【0022】

また、同じく、別の機器のアプリケーションが実行中の状態から、何らかの理由でアプリケーションを実行できない状態に変化した場合、そのアプリケーションの実行状態を引き継いだ上で、別の機器に代わって自機器が保持する同一のアプリケーションを自動的に実行することを特徴とする。

【0023】

さらに本発明の機器は、別の機器上に存在するアプリケーションと対応する、機器が保持する別のアプリケーションを検索し特定する、代替アプリ判定手段を備える。

【0024】

別の機器上で実行不能になったアプリケーションの、代替アプリケーションを機器が保持している場合には、別の機器に対して何らかの形でアプリケーションの実行要求が入ったにも関わらず、別の機器がそのアプリケーションを実行できない状態にある場合、別の機器に代わって自機器が保持する代替アプリケーションを自動的に実行することを特徴とする。

【0025】

また、同じく、別の機器のアプリケーションが実行中の状態から、何らかの理由でアプリケーションを実行できない状態に変化した場合、そのアプリケーションの実行状態を引き継いだ上で、別の機器に代わって自機器が保持する代替のアプリケーションを自動的に実行することを特徴とする。

【発明の効果】

【0026】

ある機器が備えているアプリケーションの実行が不可能になり、サービスが不能の状態に陥った場合も、本発明の装置が備える、アプリ同一性判定手段と、それを用いたアプリケーションの自動実行の仕組みによって、機器が保持するアプリケーション間で予め定められた特別な登録がなくとも、同じサービスを自動的に他の機器で実行したり、中断されたサービスの情報を他の機器で引き継いで利用したりすることができる。これによりサービス実行の安定性を大きく高めることができる。

【0027】

また、もし他の機器が同一のアプリケーションが保持していない場合でも、その次善策として、本発明の装置が備える、代替アプリ判定手段と、それを用いた代替アプリケーションの自動実行の仕組みによって、実行不能に陥ったサービスと類似のサービス、あるいは実行不能に陥ったサービスを補佐するサービスを、自動的に他の機器で実行したり、中断されたサービスの情報を他の機器で引き継いで利用したりすることができる。これによりサービス実行の柔軟性・利便性を高めることができる。

【0028】

以上の本発明の仕組みを用いることによって、センターサーバのような集中管理方式を前提としていないネットワークの機器でも、サービスの高い安定性を確保することができるようになる。

【発明を実施するための最良の形態】

【0029】

(実施の形態1)

以下、本発明の実施の形態を説明する。図1は、本実施の形態の装置の全体の概要を示

すものである。

【0030】

本発明の装置は、少なくとも2つ以上の電子機器がネットワークを介して接続された環境下において実行されるものである。ネットワークは、接続された機器間でデータの通信が可能なものであれば実施形態および通信の規格は何でもよい。インターネットのようなパブリックなネットワークであっても、LANのように限られたドメインの中で閉じられたネットワークであっても良いし、その両者の組合せからなっても良い。

【0031】

本実施の形態では説明を簡単にするために、図1に示す通りに、ネットワーク100を介して電子機器10と電子機器20が接続されているものを例に説明を行う。

【0032】

電子機器10は、アプリケーションプログラム101（以下アプリケーションと称す）を解釈し、実行するためのアプリ実行手段102と、アプリ認識手段103と、アプリ同一性判定手段104と、アプリ使用不能検出手段105と、アプリ起動要求検出手段106と、アプリ実行状態取得手段107と、代替アプリ判定手段108を備えたものである。

【0033】

電子機器20は、アプリケーションプログラム201（以下アプリケーションと称す）を解釈し、実行するためのアプリ実行手段202を備えたものである。

【0034】

図1では電子機器10と電子機器20のみが、ネットワーク100に接続されている構成であるが、もちろん実施の際には、電子機器10の構成を持つ電子機器が最低1つ以上と、電子機器20の構成を持つ電子機器が最低1つ以上備わっている構成であればよい。例えば電子機器20と同様の構成を持つ他の電子機器が、2つ以上ネットワーク100に接続されていても構わない。

【0035】

アプリ実行手段102、202は、アプリケーション101、201を解釈実行することが可能なものであれば何でも良く、例えばJ A V A（R）仮想マシン、.NET実行環境、K D D I社製の携帯電話にて採用されているプログラム配信実行環境であるB R E Wなど、既存の環境によって実現することができる。

【0036】

本実施の形態にあたっては、アプリ実行手段102、202をJ A V A（R）仮想マシン環境、アプリケーション101、201はJ A V A（R）アプリケーションとしたものを例に説明を行う。

【0037】

アプリケーション101、201は、アプリケーションプログラムの本体（J A V A（R）アプリケーションにおいてはJ A Rファイル）以外に、アプリケーションの属性を示す固有の情報（以下、アプリ属性情報と称する）を保持する。

【0038】

アプリ属性情報とは、アプリケーションの名称、アプリケーションの取得元、アプリケーションを動作させるために必要なプロファイルの情報、など、アプリケーションの特徴を示した情報のことである。

【0039】

例えば、J A V A（R）のモバイル機器向けの標準プロファイルであるM I D Pプロファイルに準拠したJ A V A（R）アプリケーションでは、アプリ属性情報はJ A Dファイルに記述されることによって提供される。しかしJ A Dファイルのようなアプリ属性情報が提供されない場合でも、アプリケーションのサイズなどの情報は、J A V A（R）アプリケーションの本体であるJ A Rファイルを解析することで獲得することができる。

【0040】

図2に、アプリ属性情報の実現例をテーブル形式で示す。それぞれのエントリにはアプ

리케이션の属性を示す情報が記録される。図に示したそれぞれのエントリについて簡単に説明する。

【0041】

アプリケーション名には、アプリケーションに与えられた名称が記述される。

【0042】

アプリケーションの取得元には、電子機器に対するアプリケーションの配布元が記述される。図ではURL形式で記述しているが、もちろん他の情報であっても良い。

【0043】

メインクラス名には、アプリケーションを開始する際に最初に呼ばれるメソッドであるmainメソッドを実装したクラス名が記述される。

【0044】

必要プロファイルには、アプリケーションを動作させるためにアプリ実行手段102が備えていることが必要な、コンフィグレーション・プロファイル・ライブラリ等の情報が記述される。

【0045】

必要リソースには、アプリケーションを動作させるために必要な資源量が記述される。

【0046】

最終更新日には、アプリケーションを最後に更新した日時が記述される。

【0047】

バージョン番号には、アプリケーションに与えられたバージョン管理番号が記述される。

【0048】

アプリケーションサイズには、アプリケーションの本体（JAVA（R）の場合、JARファイル）のサイズが記述される。

【0049】

アプリケーション作成者には、アプリケーションの作成者を示すための識別情報が記述される。図では作成者の名称を示す文字列で記述しているが、もちろん他の情報であっても良い。

【0050】

代替可能アプリケーションには、他のアプリケーションを特定するための情報が記述される。もしアプリケーションが実行中に中断されてしまった場合、アプリケーションの実行状態を引き継いで、本エントリに記録されたアプリケーションが、代わりにサービスを継続可能であることを示す。本エントリがどのように使用されるかに関しては後述する。

【0051】

被代替可能アプリケーションには、他のアプリケーションを特定するための情報が記述される。記述された情報に当てはまるアプリケーションがもし実行中に中断されてしまった場合、本アプリケーションがそのアプリケーションの実行状態を引き継いでサービスを継続可能であることを示す。本エントリがどのように使用されるかに関しては後述する。

【0052】

図2に示したものはアプリ属性情報の実現例の一つである。図2の示したテーブルのエントリの一部がなくても、あるいは他の属性の情報を示すエントリを含んでいてもよい。

【0053】

アプリ実行手段102、202は、あるアプリケーションを新たに起動できるかどうかの可否判定を行うことができる能力を有するものとする。

【0054】

このため、アプリ実行手段102、202は、自身の機器内において使用することができるハードウェアの空き資源量の情報を取得できるものとする。空き資源量とは、例えば電子機器のメモリ（図示せず）におけるアプリケーション実行用に割り当てられた容量のうち、どの動作中のアプリケーションからも占有されておらず、新たなアプリケーションを動作させるために使用可能な容量などのことである。

【 0 0 5 5 】

これは、電子機器内でアプリ実行手段が使用することができる資源量を監視することによって実現することができる。アプリ実行手段を J A V A (R) 仮想マシンで実現する場合には、J A V A (R) 仮想マシンの保持するヒープメモリ全体の使用量・未使用量、アプリケーション毎のヒープメモリの使用量など、ヒープメモリの使用状況を記録するレジスタ（図示せず）を設け、必要に応じて（例えばアプリケーション起動前、アプリケーションの動作中、アプリケーションの開放時など）レジスタの記録情報を調べるレジスタ記録情報調査手段（図示せず）を設けることによってアプリ実行手段が使用できる資源量の監視を実現することができる。

【 0 0 5 6 】

アプリケーション 1 0 1、2 0 1 が、アプリ属性情報として「アプリケーションを動作させるために必要な資源量」の情報を保持していれば、アプリ実行手段 1 0 2、2 0 2 は、自身の機器内において使用することができるハードウェアの空き資源量の情報から、アプリケーション 1 0 1、2 0 1 を動作可能であるかどうかの可否判定を行うことができる。

【 0 0 5 7 】

また、アプリ実行手段 1 0 2、2 0 2 は、アプリケーションの実行中に、そのアプリケーションの途中の実行状態を示す情報（以下、アプリ実行状態情報と称す）を出力することができる能力を有するものとする。

【 0 0 5 8 】

アプリ実行状態情報の具体例として、将棋のゲームを行うアプリケーションを考える。アプリケーションは、ゲームを始めたら通常はゲームが終わるまで休みなくアプリケーションの実行が続くが、もしゲームの途中で中断して後で再開したい場合には、中断する時点でのプレイヤーの駒の位置や持ち駒の情報を記録する必要がある。このようにアプリケーションの実行の途中の状態を示し、仮にアプリケーションの実行を中断したとしても後で再開するための情報を含んだものが、アプリ実行状態情報である。

【 0 0 5 9 】

J A V A (R) アプリケーションの実行においては、実行中のオブジェクトを保存することで、アプリ実行状態情報を記録することができる。

【 0 0 6 0 】

実行中のオブジェクトの状態を出力する手段として、J A V A (R) ではシリアライズがサポートされている。シリアライズとは、オブジェクトが持つ属性の値や構造を、バイト列で扱える形に変換することである。すなわちシリアライズによって、複雑な構造を持った実行途中のアプリケーションの状態を示すデータなどを、バイト列のストリームやファイル形式で出力し、アプリ実行状態情報とすることができる。

【 0 0 6 1 】

もちろん、オブジェクトをバイト列に変換するシリアライズの機能を用いなくとも、アプリケーションが他のフォーマットによってアプリ実行状態情報を出力する手段を保持していても良い。例えば多くのゲームアプリケーションは、ゲームの途中経過（アプリ実行状態情報）を何らかのフォーマットで記述する能力を、アプリケーション自体が保持している。アプリ実行状態情報は、例えば X M L や H T M L など、汎用性を持ったフォーマットで記述されていてもよいし、アプリケーション独自で定義したフォーマットを用いていても良い。

【 0 0 6 2 】

アプリ認識手段 1 0 3 は、ネットワーク 1 0 0 を通じて接続された他の電子機器が保持するアプリケーションの特徴を獲得する能力を有する。この獲得処理は、他の電子機器が保持するアプリケーションのアプリ属性情報を、ネットワーク 1 0 0 を通じて取得することで実現することができる。

【 0 0 6 3 】

アプリ同一性判定手段 1 0 4 は、アプリ認識手段 1 0 3 によって認識された他の機器の

アプリケーションのアプリ属性情報と、自機器の保持するアプリケーションのアプリ属性情報とを比較して、同一性の判定を行う。

【0064】

アプリケーションの同一性判定処理の実現例を図3に示す。ここでは自機器のアプリケーションをアプリケーション101、他機器の比較対象のアプリケーションをアプリケーション201として説明を行う。

【0065】

処理301は、アプリケーション101のアプリ属性情報と、アプリケーション201のアプリ属性情報のそれぞれから「アプリケーションの名称」を取得する。

【0066】

処理302は、処理301の処理で情報が正常に取得できたかどうかを判定する。アプリケーション101あるいはアプリケーション201のいずれかから、比較対照の情報（ここでは「アプリケーションの名称」）を取得できなかった場合、両アプリケーションの同一性を確認できなかったものとして、処理308に移行する。

【0067】

処理303は、処理301にて取得した両アプリ属性情報を比較する。比較の結果、固有情報が異なっていれば、両アプリケーションの同一性を確認できなかったものとして、処理308に移行する。

【0068】

処理304は、アプリケーション101のアプリ属性情報と、アプリケーション201のアプリ属性情報のそれぞれから、「アプリケーションの作成者」を取得する。

【0069】

処理305は、処理304の処理で情報が正常に取得できたかどうかを判定する。アプリケーション101あるいはアプリケーション201のいずれかから、比較対照の情報（ここでは「アプリケーションの作成者」）を取得できなかった場合、両アプリケーションの同一性を確認できなかったものとして、処理308に移行する。

【0070】

処理306は、処理301にて取得した両アプリ属性情報を比較する。比較の結果、固有情報が異なっていれば、両アプリケーションの同一性を確認できなかったものとして、処理308に移行する。

【0071】

処理307は、アプリケーション101とアプリケーション201が同一のものであると判定を下す。処理307に移行せずに、処理308に移行した場合には、アプリケーション101とアプリケーション201が同一のものではないと判定を下す。

【0072】

以上がアプリケーションの同一性判定処理の実現例である。この実現例では、アプリケーションの同一性の判定に「アプリケーションの名称」と「アプリケーションの作成者」という2つのアプリ属性情報を使用しているが、必ずしもこれらを使用する必要はない。他のアプリ属性情報の組み合わせで比較して同一性判定を行っても良いし、比較に使用するアプリ属性情報の種類は幾つでもよい。

【0073】

図3に示した処理はアプリケーション101とアプリケーション201のそれぞれ1対1の比較のみを説明したが、電子機器10が複数のアプリケーションを保持している場合など、比較対照となるアプリケーションが複数ある場合には、それぞれのアプリケーションの考えられる組み合わせ全てに対してこの処理を行う。

【0074】

アプリ同一性判定手段104は、同一と判定したアプリケーションの記録を保持する。ここでは情報をテーブルに記録した実現例を図4に示す。

【0075】

図4中の(1)はテーブル本体である。テーブルのエントリは、他の機器が保持するア

アプリケーションを示す情報（テーブルの左列）と、そのアプリケーションと同一と判定された、自機器が保持するアプリケーションを示す情報（テーブルの右列）の組から構成される。

【 0 0 7 6 】

図 4 中の（2）は、他の機器が保持するアプリケーションを示す情報である。この実現例では、アプリケーションを特定する情報として「アプリケーションが保存されている機器の識別情報」「アプリケーションの名称」「アプリケーションのサイズ」の組み合わせを使用している。もちろん、他の機器が保持するアプリケーションを特定することができる情報であれば、これ以外の情報が使用されていてもよい。

【 0 0 7 7 】

図 4 中の（3）は、自機器が保持するアプリケーションを示す情報である。この実現例では、アプリケーションを特定する情報として「アプリケーションの名称」「アプリケーションのサイズ」の組み合わせを使用している。もちろん、自機器が保持するアプリケーションを特定することができる情報であれば、これ以外の情報が使用されていてもよい。

【 0 0 7 8 】

アプリ使用不能検出手段 1 0 5 は、ネットワーク 1 0 0 を通じて接続された他の電子機器の保持するアプリケーション 2 0 1 が、実行可能な状態にあるか否かを検出する。アプリケーションが実行可能な状態にない、とは大きく以下の 2 つの場合を想定している。

【 0 0 7 9 】

（A）ネットワーク 1 0 0 を通じて接続された他の電子機器 2 0 が、通電状態を保っていない

（B）ネットワーク 1 0 0 を通じて接続された他の電子機器 2 0 の保持する、アプリ実行手段 2 0 2 の資源制約の問題から、アプリケーションを実行することができない

アプリ使用不能検出手段 1 0 5 は、上記（A）と（B）の状態に陥っていないかどうかを、他機器のアプリケーションについて監視を行うことで判定を行う。アプリ使用不能検出手段の実現例を図 5 に示す。

【 0 0 8 0 】

処理 4 0 1 と処理 4 0 2 では、上記（A）の要素について判定を行っている。処理 4 0 1 で電子機器 2 0 の通電状態を獲得するが、これは従来の技術を用いて検出することが可能である。

【 0 0 8 1 】

例えば家電ネットワークの接続規格の一つである U P n P の仕様では、ネットワーク接続された通電状態にある電子機器は、一定時間ごとにネットワーク接続されている機器に信号を発することになっている。もし電子機器から定期的な信号が発信されない場合には、当該電子機器は通電状態が切断されたものと見なされる。処理 4 0 1 で電子機器 2 0 の通電状態が切断されていると検出された場合には、処理 4 0 2 から処理 4 0 9 へ移行し、電子機器 2 0 の保持する全てのアプリケーションは実行可能な状態にないと判定される。

【 0 0 8 2 】

処理 4 0 3 から 4 0 8 にかけては、電子機器 2 0 の保持する全てのアプリケーションごとに上記（B）の要素についての判定を行っている。

【 0 0 8 3 】

処理 4 0 4 では、電子機器 2 0 のアプリ実行手段 2 0 2 から、アプリケーションの実行が可能であるかどうかの情報を獲得する。先のアプリ実行手段の説明にて記載した通り、アプリケーション 2 0 1 が、アプリケーション固有の情報として「アプリケーションを動作させるために必要な資源量」の情報を保持していれば、アプリ実行手段 2 0 2 は、自身の機器内において使用することができるハードウェアの空き資源量の情報から、アプリケーション 2 0 1 を動作可能であるかどうかの可否を判断することができる。処理 4 0 4 はアプリ実行手段 2 0 2 に対して、この可否の問い合わせを行い、結果を獲得する。

【 0 0 8 4 】

処理 4 0 5 によって、アプリ実行手段 2 0 2 から返された結果を判定し、アプリケーション

ョンが実行可能状態にある場合は、処理 4 0 6 に、実行可能な状態にない場合は処理 4 0 7 に移行し、電子機器 2 0 が保持しているアプリケーションそれぞれの判定結果を得る。

【0 0 8 5】

アプリ起動要求検出手段 1 0 6 は、別の機器上に存在するアプリケーションに対しての起動要求が発生した場合これを検出し、どの機器が保持している、どのアプリケーションに対して起動要求が行われたか、を特定する。

【0 0 8 6】

アプリケーションに対して発生する起動要求の種類には、本発明では大きく以下の 2 通りを想定している。

【0 0 8 7】

(I) 他の機器からネットワーク 1 0 0 を通じ、機器とアプリケーションを特定する情報を伴って、起動要求が送信される場合。

【0 0 8 8】

(I I) ネットワークを通さずに電子機器が保持するアプリケーションが起動される場合。電子機器に対して直接ユーザがアプリケーションの起動を指定する、あるいは電子機器のタイマなど内部的な条件をトリガーして起動する、など。

【0 0 8 9】

電子機器 2 0 のアプリケーション 2 0 1 に対して起動要求が発生したことを、アプリ起動要求検出手段 1 0 6 が検出するための実現例としては、以下の 2 通りが考えられる。

【0 0 9 0】

(i) アプリ起動要求検出手段 1 0 6 はネットワーク 1 0 0 の通信状態を監視する。もしネットワーク 1 0 0 に接続された何らかの機器から、電子機器 2 0 のアプリケーション 2 0 1 に対して起動要求が発せられた場合、アプリ起動要求検出手段 1 0 6 はこれを検知する。本手段によって上記 (I) の起動要求を検出することが可能になる。

【0 0 9 1】

(i i) 電子機器 2 0 の機器内部でアプリケーション 2 0 1 の起動要求が発生したことを、電子機器 2 0 に通知して貰う。電子機器 2 0 のアプリ実行手段 2 0 2 はアプリケーションの起動要求が発生した場合、当然これを検出することができる。アプリ起動要求検出手段 1 0 6 は、電子機器 2 0 で起動要求が発生したことと、どのアプリケーションに対して要求が行われたかの情報を、アプリ実行手段 2 0 2 に問い合わせを行うことで検知することができる。本手段によって上記 (I I) の起動要求を検出することが可能になる。

【0 0 9 2】

アプリ起動検出手段 1 0 6 は、(i) と (i i) の両方もしくはいずれかの仕組みを備えることで、他の機器の保持するアプリケーションに対する起動要求を検出することができる。

【0 0 9 3】

アプリ実行状態取得手段 1 0 7 は、電子機器 2 0 のアプリ実行手段 2 0 2 にて生成されるアプリ実行状態情報を取得する。先に説明した通りアプリ実行状態情報は、アプリケーションの実行過程において、シリアルライズ等の技術を用いてアプリ実行手段 2 0 2 によって生成される。これを取得し解析することによって、電子機器 1 0 は電子機器 2 0 の上で実行されているアプリケーションの実行の途中の状態を知ることが可能である。

【0 0 9 4】

代替アプリ判定手段 1 0 8 は、アプリ認識手段 1 0 3 によって認識されたアプリケーションの代替アプリケーションを、自機器の保持するアプリケーションの中から検索する。あるアプリケーションが実現するサービスの類似のサービス、あるいはサービスを補佐するサービスを実現するアプリケーションのことを、ここでは代替アプリケーションと呼んでいる。

【0 0 9 5】

電子機器 2 0 が保持するアプリケーションと、電子機器 1 0 が保持するその代替アプリ

ケーションの関連付けの処理は、本発明では以下のいずれかの手段を用いて行われることを想定している。

【0096】

(a) 電子機器 20 が保持するアプリケーションがアプリ属性情報として、自アプリケーションの代替アプリケーションになることができる他アプリケーションの条件を保持している。図 2 にて示した「代替可能アプリケーション」の情報がこれに該当する。代替アプリ判定手段 108 はこれを解読し、自機器内に条件に合致するアプリケーションがあるか否かを検出する。

【0097】

(b) 電子機器 10 が保持するアプリケーションがアプリ属性情報として、自アプリケーションが他のアプリケーションの代替アプリケーションになるための条件を保持する。図 2 にて示した「被代替可能アプリケーション」の情報がこれに該当する。代替アプリ判定手段 108 はこれを解読し、条件に合致するアプリケーションが他の電子機器に存在するかどうかを検出する。

【0098】

(c) あらかじめ電子機器 10 に組み込まれた情報として、テーブルに記録されている。

【0099】

(d) ユーザによって関連付けが追加される。この場合の代替アプリケーションの関連付けの実現例を図 6 に示す。

【0100】

代替アプリ判定手段 108 は、図 6 中の (1) (2) (3) (4) のようなユーザインタフェースを備え、ユーザはこれを用いて代替アプリケーションの登録を行うことができる。

【0101】

図 6 (1) の画面は、電子機器 20 が保持するアプリケーションの一覧を表示する。これは電子機器 20 のアプリ認識手段 103 がネットワーク 100 を通じて、電子機器 20 が保持するアプリケーション固有の情報を取得することによって実現することができる。この画面でユーザはアプリケーションを選択・指定することで、図 6 (2) の画面に移行する。

【0102】

図 6 (2) の画面は、電子機器 10 が保持するアプリケーションの一覧を表示している。これは電子機器 10 が保持するアプリケーション固有の情報を取得することによって実現することができる。この画面でユーザは、先の図 6 (1) で指定したアプリケーションの代替アプリケーションを選択・指定することで、図 6 (3) の画面に移行する。

【0103】

図 6 (3) の画面は、代替アプリケーションの関連付けの確認を行う。ここでユーザは "Y e s" を指定することで、図 6 (4) の画面に移行し、関連付けが確定する。

【0104】

代替アプリ判定手段 108 は、アプリケーションの関連付けの情報を何らかの形式で保持する。ここでは情報をテーブルにとして記録した実現例を図 7 に示す。

【0105】

図 7 (1) はテーブル本体である。テーブルのエントリは、他の機器（電子機器 20）が保持するアプリケーションを示す情報と、そのアプリケーションに対応した、自機器（電子機器 10）が保持する代替アプリケーションを示す情報の組から構成される。

【0106】

図 7 (2) は、他の機器（電子機器 20）が保持するアプリケーションを示す情報である。この実現例では、アプリケーションを特定する情報として「アプリケーションが保存されている機器の識別情報」「アプリケーションの名称」「アプリケーションのサイズ」の組み合わせを使用している。もちろん、他の機器（電子機器 20）が保持するアプリケ

ーションを特定することができる情報であれば、これ以外の情報が使用されていてもよい。

【0 1 0 7】

図 7 (3) は、自機器 (電子機器 1 0) が保持する代替アプリケーションを示す情報である。この実現例では、アプリケーションを特定する情報として「アプリケーションの名称」「アプリケーションのサイズ」の組み合わせを使用している。もちろん、自機器 (電子機器 1 0) が保持するアプリケーションを特定することができる情報であれば、これ以外の情報が使用されていてもよい。

【0 1 0 8】

以上、図 1 に示された本実施の形態の構成について説明した。

【0 1 0 9】

次に、本実施の形態における、全体の処理の流れについて説明する。本発明では、上記の構成の組み合わせを用いて、4 種類の処理の流れを行うことを想定している。それぞれの実現例を以下に記す。

【0 1 1 0】

図 8 は、請求項 1 にて記載した電子機器 1 0 の全体の処理の流れを示している。

【0 1 1 1】

電子機器 1 0 は、定期的に、アプリ同一性判定手段 1 0 4 と、アプリ使用不能検出手段 1 0 5 の判定処理を実行している。

【0 1 1 2】

定期的に判定処理が行われることを保証するために、図 8 では電子機器 1 0 内にタイマの存在を仮定しており、タイマに従い定められたタイミングで判定処理を周期的に行う実装にしている (処理 8 0 0、処理 8 0 1、処理 8 0 2)。

【0 1 1 3】

アプリ同一性判定手段 1 0 4、アプリ使用不能検出手段 1 0 5 の判定処理は特に順序立てて行われる必要はなく、定期的に処理が実行されてさえいれば、処理の周期は揃っていても構わない。例えばアプリ同一性判定手段 1 0 4 は 3 0 0 s 毎に実行し、アプリ使用不能検出手段 1 0 5 は 5 s 毎に実行する、などそれぞれ固有の実行タイミングを保持していてもよい。

【0 1 1 4】

処理 8 0 3 では、アプリ同一性判定手段 1 0 4 による処理を行っている。他の電子機器が保持するアプリケーションを対象にアプリ同一性判定手段の処理を行い、同一とみなされるアプリケーションが発見されたら、その組み合わせをテーブルとして記録する。

【0 1 1 5】

処理 8 0 4 では、アプリ使用不能検出手段 1 0 5 による判定処理を行っている。他の電子機器が保持するアプリケーションの中に、実行不能と検出されたアプリケーションが発見された場合、処理 8 0 5 に移行する。

【0 1 1 6】

処理 8 0 5 では、アプリ起動要求検出手段 1 0 6 を用いた判定処理を行っている、処理 8 0 4 で実行不能と検出されたアプリケーションに対して起動要求が確認された場合、処理 8 0 6 に移行する。

【0 1 1 7】

処理 8 0 6 では、アプリ同一性判定手段 1 0 4 により作成されたテーブルを参照し、処理 8 0 5 で起動要求のあったアプリケーションと同一のアプリケーションを電子機器 1 0 が保持しているかどうかを判定する。保持している場合には処理 8 0 7 に移行する。

【0 1 1 8】

処理 8 0 7 では、処理 8 0 6 にて同一と確認されたアプリケーションを起動し、アプリケーションの実行を開始する。

【0 1 1 9】

この処理の流れを利用した具体的なサービスの例を挙げる。

【0 1 2 0】

ホームサーバ1は定期的に、ネットワーク接続された別の機器のデータを自機器にバックアップしコピーするサービスを備えているとする。ホームサーバ1はバックアップを実行するアプリケーション（バックアップアプリ1）を保持し、内部のタイマによって起動要求がかかることによってサービスを実現する。

【0 1 2 1】

このホームサーバ1のバックアップアプリ1が、実行不能な状態に陥ったとする。

【0 1 2 2】

ここで、ホームサーバ1にネットワーク接続されたホームサーバ2は、バックアップアプリ1と同一のバックアップアプリ2を保持している。ホームサーバ2は、バックアップアプリ1が実行不能な状態に陥ったことを検出し、かつ、ホームサーバ1の内部のタイマによりバックアップアプリ1に起動要求が掛けられたことを検出すると、ホームサーバ2はバックアップアプリ2を起動する。こうしてバックアップアプリ1の起動が不可能なホームサーバ1に代わって、ホームサーバ2によるサービスが実現される。

【0 1 2 3】

図9は、請求項2にて記載した電子機器10の全体の処理の流れを示している。

【0 1 2 4】

電子機器10は、定期的に、アプリ同一性判定手段104と、アプリ使用不能検出手段105と、アプリ実行状態取得手段107の判定処理を実行している。

【0 1 2 5】

定期的に判定処理が行われることを保証するために、図9では電子機器10内にタイマの存在を仮定しており、タイマに従い定められたタイミングで判定処理を周期的に行う実装にしている（処理900、処理901、処理902、処理903）。

【0 1 2 6】

アプリ同一性判定手段104、アプリ使用不能検出手段105、アプリ実行状態取得手段107の判定処理は特に順序立てて行われる必要はなく、定期的に処理が実行されてさえいれば、処理の周期は揃っていなくとも構わない。

【0 1 2 7】

処理904では、アプリ同一性判定手段104による処理を行っている。他の電子機器が保持するアプリケーションを対象にアプリ同一性判定手段の処理を行い、同一とみなされるアプリケーションが発見されたら、その組み合わせをテーブルとして記録する。

【0 1 2 8】

処理905では、アプリ実行状態取得手段による処理を行っている。他の電子機器上で実行されているアプリケーションのアプリ実行状態情報をここで取得する。

【0 1 2 9】

処理906では、アプリ使用不能検出手段105による判定処理を行っている。他の電子機器が保持するアプリケーションの中に、実行不能と検出されたアプリケーションが発見された場合、処理907に移行する。

【0 1 3 0】

処理907では、処理906で実行不能と検出されたアプリケーションのアプリ実行状態情報を電子機器10が保持するか否かを判定する。保持している場合には処理908に移行する。

【0 1 3 1】

処理908では、アプリ同一性判定手段104により作成されたテーブルを参照し、処理906で実行不能と検出されたアプリケーションと同一のアプリケーションを電子機器10が保持しているかどうかを判定する。保持している場合には処理909に移行する。

【0 1 3 2】

処理909では、処理908にて同一と確認されたアプリケーションの実行を開始する。この際には、処理907で保持していると判定された、アプリ実行状態情報を利用し、その実行状態を引き継ぐ形でアプリケーションの実行を開始する。

【0 1 3 3】

この処理の流れを利用した具体的なサービスの例を挙げる。

【0 1 3 4】

A V 機器 1 は放送録画を行うサービスを備えているとする。A V 機器 1 は放送の録画を実行するアプリケーション（録画アプリ 1）を保持し、サービスを実現する。

【0 1 3 5】

この A V 機器 1 の録画アプリ 1 が、放送の録画中に突如、実行不能な状態に陥り、サービスが中断されてしまったとする。

【0 1 3 6】

ここで、A V 機器 1 にネットワーク接続された A V 機器 2 は、録画アプリ 1 と同一の録画アプリ 2 を保持している。A V 機器 2 は A V 機器 1 上で実行されていた録画アプリ 1 の、どのチャンネルをどのような設定で録画したか、といったアプリ実行状態情報を取得している。

【0 1 3 7】

A V 機器 2 は、録画アプリ 1 が実行不能な状態に陥ったことを検出すると、A V 機器 2 は録画アプリ 2 を起動し、録画アプリ 1 のアプリ実行状態情報を引き継ぎ、録画アプリ 1 が行っていたのと同じ設定でサービスを引継ぎ、実現する。

【0 1 3 8】

図 1 0 は、請求項 3 にて記載した電子機器 1 0 の全体の処理の流れを示している。

【0 1 3 9】

電子機器 1 0 は、定期的に、代替アプリ判定手段 1 0 8 と、アプリ使用不能検出手段 1 0 5 の判定処理を実行している。

【0 1 4 0】

定期的に判定処理が行われることを保証するために、図 1 0 では電子機器 1 0 内にタイマの存在を仮定しており、タイマに従い定められたタイミングで判定処理を周期的に行う実装にしている（処理 1 0 0 0、処理 1 0 0 1、処理 1 0 0 2）。

【0 1 4 1】

代替アプリ判定手段 1 0 8、アプリ使用不能検出手段 1 0 5 の判定処理は特に順序立てて行われる必要はなく、定期的に処理が実行されてさえいれば、処理の周期は揃っていても構わない。

【0 1 4 2】

処理 1 0 0 3 では、代替アプリ判定手段 1 0 8 による処理を行っている。他の電子機器が保持するアプリケーションを対象に、代替アプリ判定手段の処理を行い、代替アプリケーションが発見されたら、その組み合わせをテーブルとして記録する。

【0 1 4 3】

処理 1 0 0 4 では、アプリ使用不能検出手段 1 0 5 による判定処理を行っている。他の電子機器が保持するアプリケーションの中に、実行不能と検出されたアプリケーションが発見された場合、処理 1 0 0 5 に移行する。

【0 1 4 4】

処理 1 0 0 5 では、アプリ起動要求検出手段 1 0 6 を用いた判定処理を行っている、処理 1 0 0 4 で実行不能と検出されたアプリケーションに対して起動要求が確認された場合、処理 1 0 0 6 に移行する。

【0 1 4 5】

処理 1 0 0 6 では、代替アプリ判定手段 1 0 8 により作成されたテーブルを参照し、処理 1 0 0 5 で起動要求のあったアプリケーションの代替アプリケーションを電子機器 1 0 が保持しているかどうかを判定する。保持している場合には処理 1 0 0 7 に移行する。

【0 1 4 6】

処理 1 0 0 7 では、処理 1 0 0 6 にて特定された代替アプリケーションを起動し、アプリケーションの実行を開始する。

【0 1 4 7】

この処理の流れを利用した具体的なサービスの例を挙げる。

【0 1 4 8】

ホームサーバは定期的に、ネットワーク接続された別の機器のデータを自機器にバックアップしコピーするサービスを備えているとする。ホームサーバはバックアップを実行するアプリケーション（バックアップアプリ）を保持し、内部のタイマによって起動要求がかかることによってサービスを実現する。

【0 1 4 9】

このホームサーバのバックアップアプリが、実行不能な状態に陥ったとする。

【0 1 5 0】

ここで、ホームサーバとネットワーク接続された携帯電話は、バックアップアプリの代替アプリケーションとして関連付けされた、ユーザ通知アプリケーションというアプリケーションを保持している。

【0 1 5 1】

携帯電話は、ホームサーバのバックアップアプリが実行不能な状態に陥ったことを検出し、かつ、ホームサーバの内部のタイマによりバックアップアプリに起動要求が掛けられたことを検出すると、ユーザ通知アプリケーションを起動する。ユーザ通知アプリケーションは、ホームサーバのバックアップが失敗したことをユーザに対して通知、警告するサービスを実施する。

【0 1 5 2】

こうして、バックアップアプリによるサービスを補佐する形での別のサービスの実施が実現できる。

【0 1 5 3】

図 1 1 は、請求項 4 にて記載した電子機器 1 0 の全体の処理の流れを示している。

【0 1 5 4】

電子機器 1 0 は、定期的に、代替アプリ判定手段 1 0 8 と、アプリ使用不能検出手段 1 0 5 と、アプリ実行状態取得手段 1 0 7 の判定処理を実行している。

【0 1 5 5】

定期的に判定処理が行われることを保証するために、図 1 1 では電子機器 1 0 内にタイマの存在を仮定しており、タイマに従い定められたタイミングで判定処理を周期的に行う実装にしている（処理 1 1 0 0、処理 1 1 0 1、処理 1 1 0 2、処理 1 1 0 3）。

【0 1 5 6】

代替アプリ判定手段 1 0 8、アプリ使用不能検出手段 1 0 5、アプリ実行状態取得手段 1 0 7 の判定処理は特に順序立てて行われる必要はなく、定期的に処理が実行されてさえいれば、処理の周期は揃っていても構わない。

【0 1 5 7】

処理 1 1 0 4 では、代替アプリ判定手段 1 0 8 による処理を行っている。他の電子機器が保持するアプリケーションを対象に、代替アプリ判定手段の処理を行い、代替アプリケーションが発見されたら、その組み合わせをテーブルとして記録する。

【0 1 5 8】

処理 1 1 0 5 では、アプリ実行状態取得手段による処理を行っている。他の電子機器上で実行されているアプリケーションのアプリ実行状態情報をここで取得する。

【0 1 5 9】

処理 1 1 0 6 では、アプリ使用不能検出手段 1 0 5 による判定処理を行っている。他の電子機器が保持するアプリケーションの中に、実行不能と検出されたアプリケーションが発見された場合、処理 1 1 0 7 に移行する。

【0 1 6 0】

処理 1 1 0 7 では、処理 1 1 0 6 で実行不能と検出されたアプリケーションのアプリ実行状態情報を電子機器 1 0 が保持するか否かを判定する。保持している場合には処理 1 1 0 8 に移行する。

【0 1 6 1】

処理 1 1 0 8 では、代替アプリ判定手段 1 0 8 により作成されたテーブルを参照し、処理 1 1 0 6 で実行不能と検出されたアプリケーションの代替アプリケーションを電子機器 1 0 が保持しているかどうかを判定する。保持している場合には処理 1 1 0 9 に移行する。

【0 1 6 2】

処理 1 1 0 9 では、処理 1 1 0 8 にて特定された代替アプリケーションの実行を開始する。この際には、処理 1 1 0 7 で保持していると判定された、アプリ実行状態情報を利用し、その実行状態を引き継ぐ形でアプリケーションの実行を開始する。

【0 1 6 3】

この処理の流れを利用した具体的なサービスの例を挙げる。

【0 1 6 4】

携帯電話は放送の視聴をユーザに対して提供するサービスを備えているとする。携帯は放送の表示を実行するアプリ（TVアプリ）を保持し、サービスを実現する。

【0 1 6 5】

この携帯電話のTVアプリが、実行中に突如、実行不能な状態に陥り、放送の視聴の提供というサービスが中断されてしまったとする。

【0 1 6 6】

ここで、携帯電話器とネットワーク接続されたAV機器は、TVアプリの代替アプリケーションとして関連付けされた、放送を自機器に記録する録画アプリというアプリケーションを保持している。AV機器は携帯電話で実行されていたTVアプリは、どのチャンネルをどのような設定で視聴されていたか、といったアプリ実行状態情報を取得している。

【0 1 6 7】

AV機器は、携帯電話のTVアプリが実行不能な状態に陥ったことを検出すると、録画アプリを起動し、TVアプリのアプリ実行状態情報を引き継ぎ、TVアプリで視聴されていたのと同じ設定でサービスを引継ぎ、放送の録画サービスを実現する。

【0 1 6 8】

携帯電話のユーザはあとでAV機器によって放送の続きを視聴することが可能である。こうしてTVアプリによるサービスを補佐する形での別のサービスの実施が実現できる。

【0 1 6 9】

以上が本発明の実施の形態の説明である。

【産業上の利用可能性】

【0 1 7 0】

以上の構成を持つ本発明を利用することで、センターサーバのように集中管理方式を前提としないネットワーク機器においても、同一のサービスを自動的に他の機器で実行したり、中断されたサービスの続きを他の機器で引き継いで実行したりといった、サービスの高い安定性を確保するための機器間の連携動作を行うことができるようになる。

【図面の簡単な説明】

【0 1 7 1】

【図 1】 本実施の形態の一例を説明するための図

【図 2】 本実施の形態のアプリケーション固有の情報（アプリ属性情報）の一例を説明するための図

【図 3】 本実施の形態のアプリ同一性判定手段の動作の一例を説明するための図

【図 4】 本実施の形態のアプリ同一性判定手段による同一アプリケーション登録の実現の一例を説明するための図

【図 5】 本実施の形態のアプリ使用不能検出手段の動作の一例を説明するための図

【図 6】 本実施の形態の代替アプリ判定手段による代替アプリケーション登録の実現の一例を説明するための図

【図 7】 本実施の形態の代替アプリ判定手段の実現の一例を説明するための図

【図 8】 本実施の形態の全体の処理の一例を説明するための図

【図 9】 本実施の形態の全体の処理の一例を説明するための図

【図 1 0】 本実施の形態の全体の処理の一例を説明するための図

【図 1 1】 本実施の形態の全体の処理の一例を説明するための図

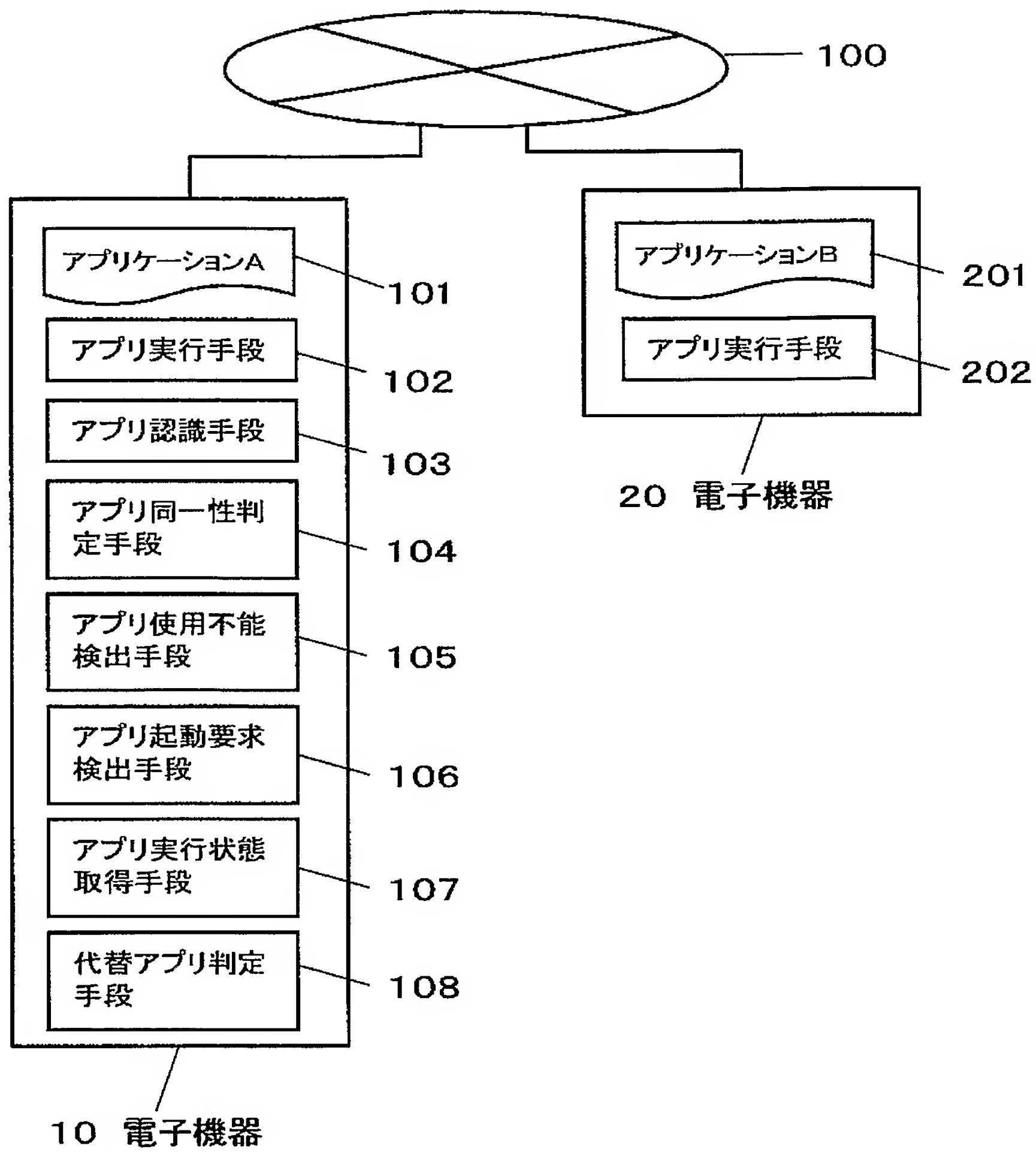
【符号の説明】

【 0 1 7 2 】

- 1 0 電子機器
- 2 0 電子機器
- 1 0 0 ネットワーク
- 1 0 1 アプリケーションプログラム
- 1 0 2 アプリ実行手段
- 1 0 3 アプリ認識手段
- 1 0 4 アプリ同一性判定手段
- 1 0 5 アプリ使用不能検出手段
- 1 0 6 アプリ起動要求検出手段
- 1 0 7 アプリ実行状態取得手段
- 1 0 8 代替アプリ判定手段

【書類名】 図面

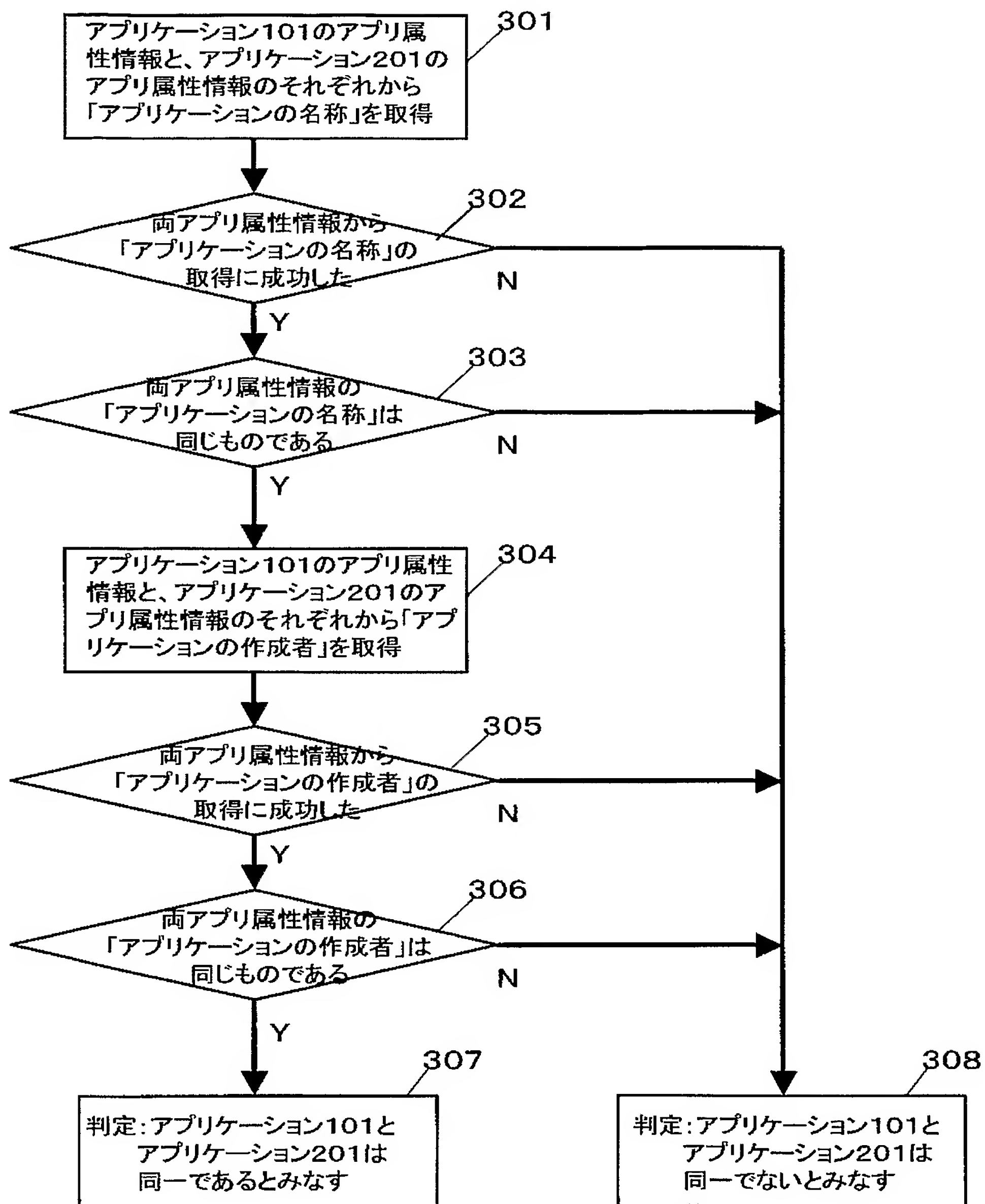
【図 1】



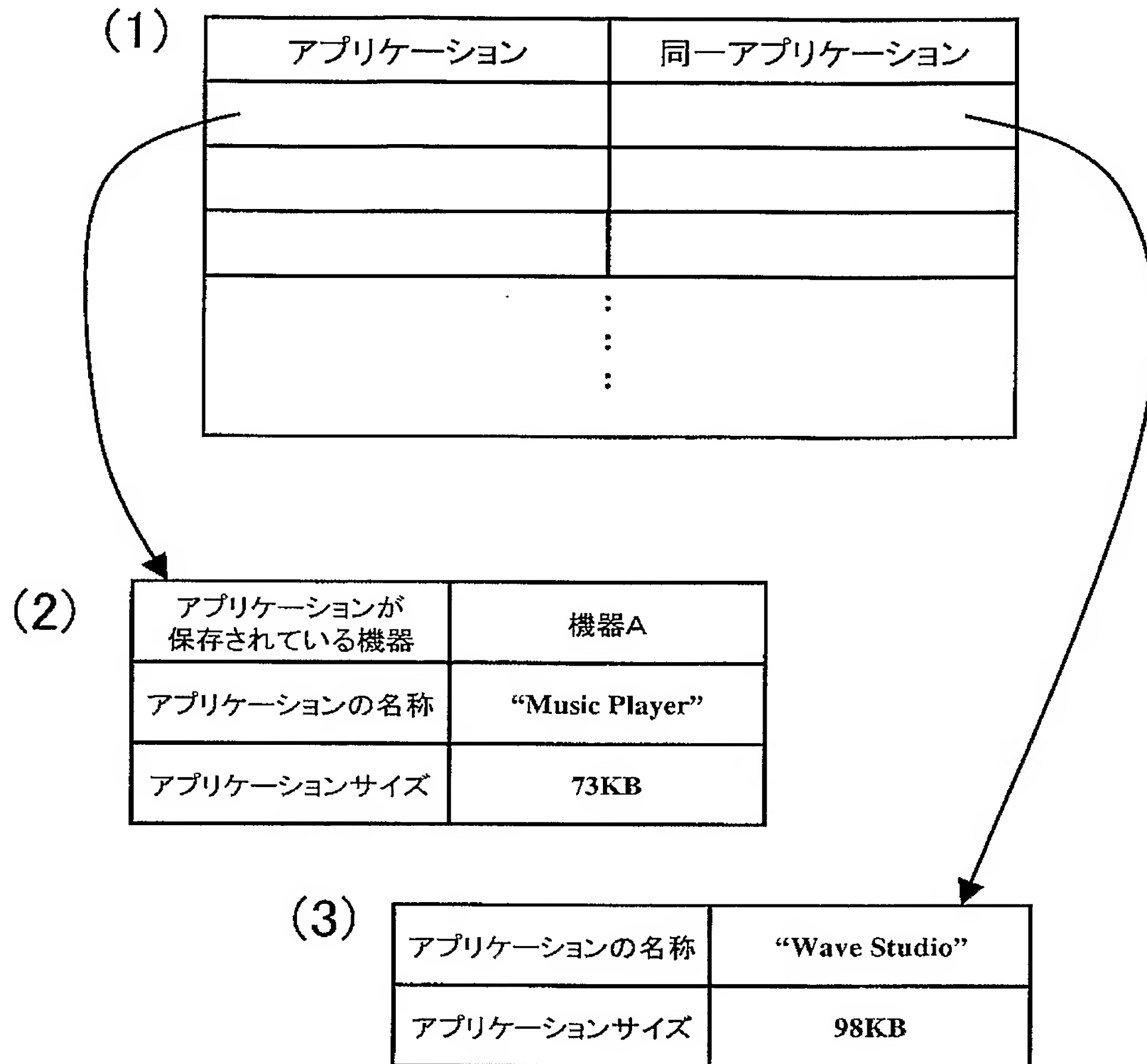
【図 2】

アプリケーション名	“Contents Recorder”
アプリケーションの取得元	http://www.distributer.xxx
メインクラス名	com.panasonic.applet
必要プロファイル	CDC／CVM
必要リソース	RAM 800KB
最終更新日	2004/01/01 18:00
バージョン番号	1.04
アプリケーションサイズ	174KB
アプリケーション作成者	“Panasonic CAC”
代替可能アプリケーション	“Media Recorder”
被代替可能アプリケーション	“Contents Mirroring”

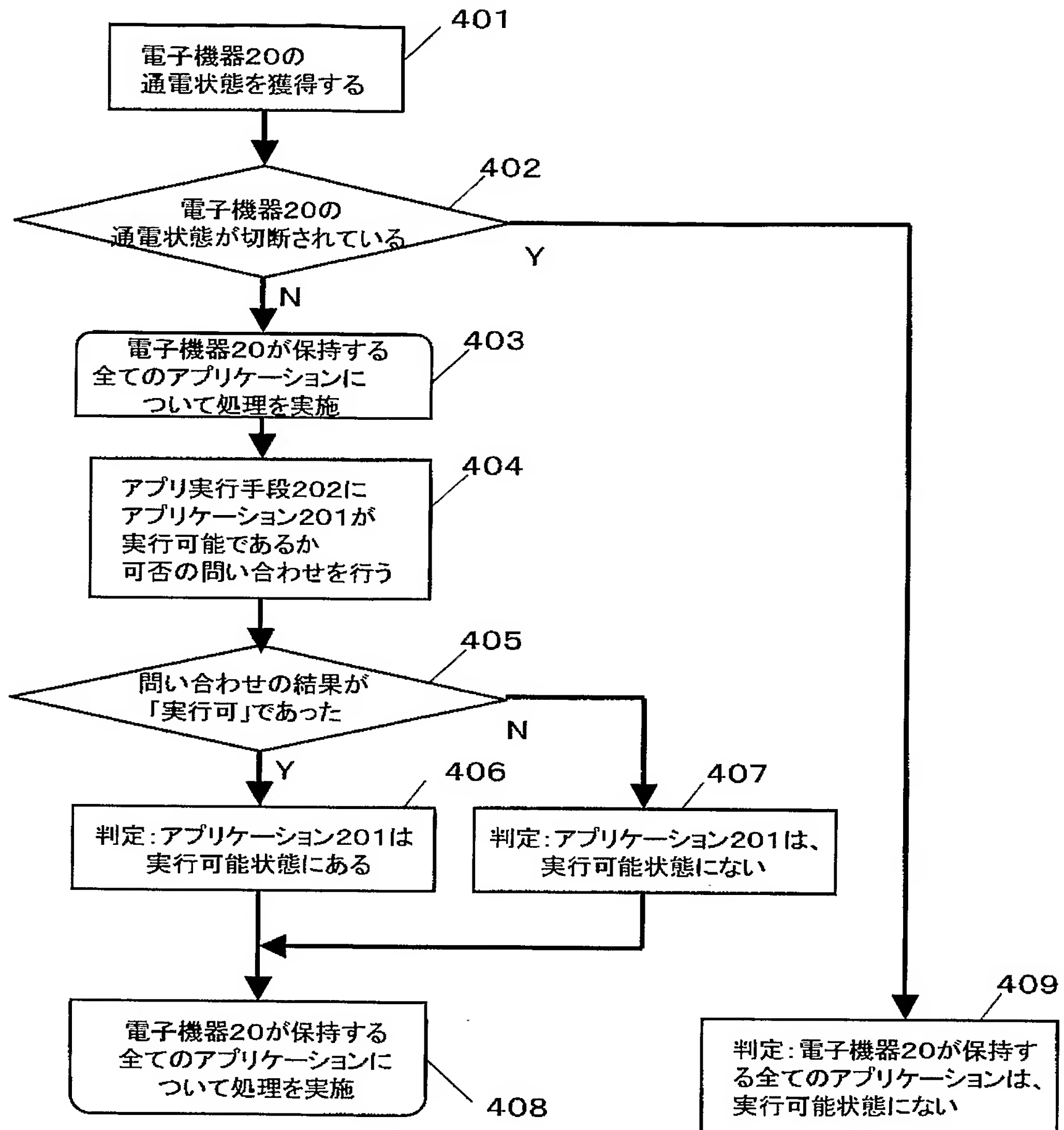
【図 3】



【図 4】



【図 5】



【図 6】

(1)

アプリケーション登録
機器20の アプリケーション一覧
Scheduler Chat Window Media Viewer News Reader

アプリケーション
“Media Viewer”を選択

(2)

代替アプリケーション登録
機器10の アプリケーション一覧
SD Reader BackUp Recorder G-Library Wave Studio

代替アプリケーションとして
“BackUp Recorder”を選択

(4)

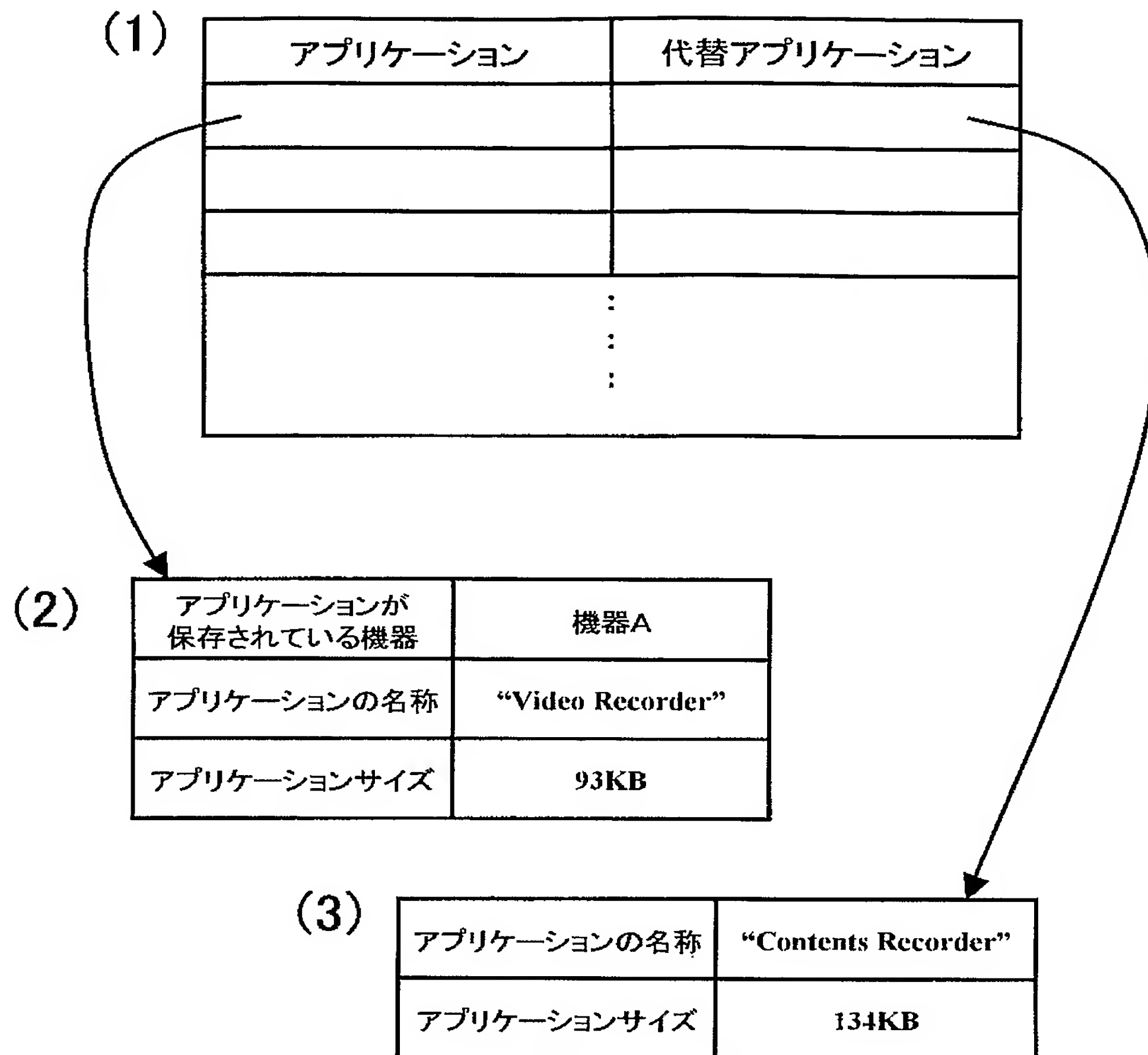
代替アプリケーションの登 録に成功しました
OK

(3)

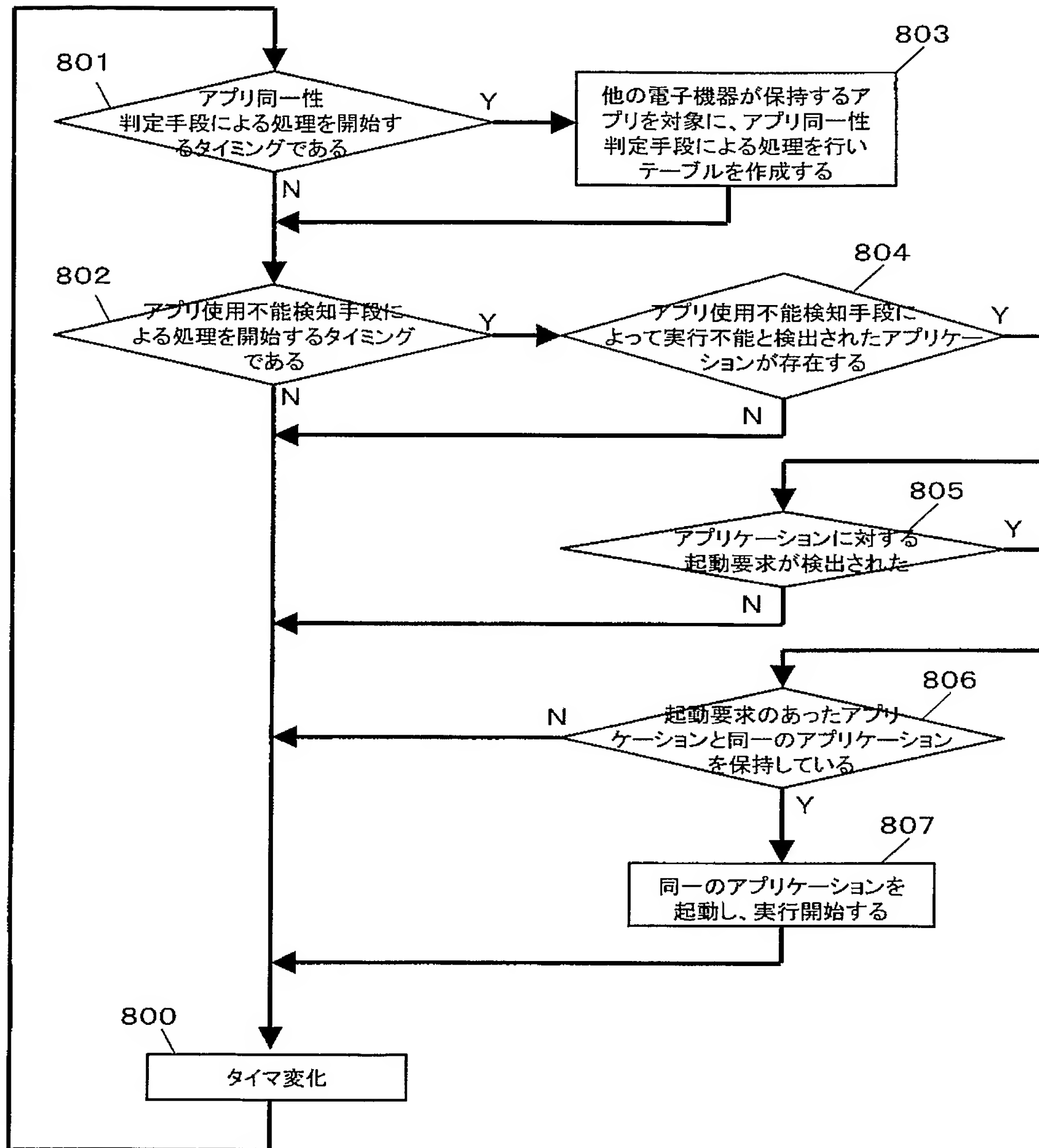
機器20のアプリケーショ ン“Media Viewer”の代替 アプリケーションとして、機 器10のアプリケーション “BackUp Recorder”に登 録します
よろしいですか？
Yes No

“Yes”を選択

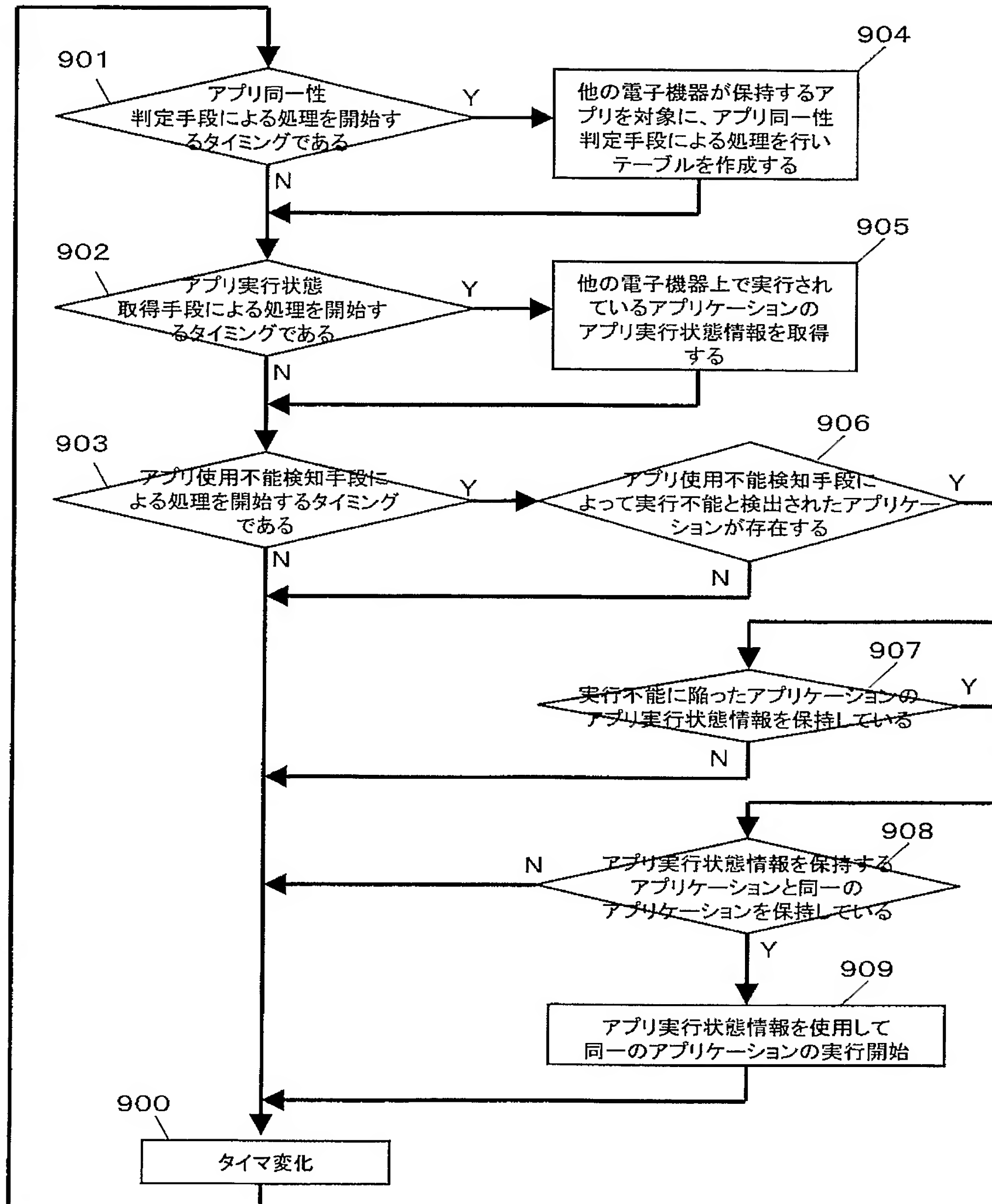
【図 7】



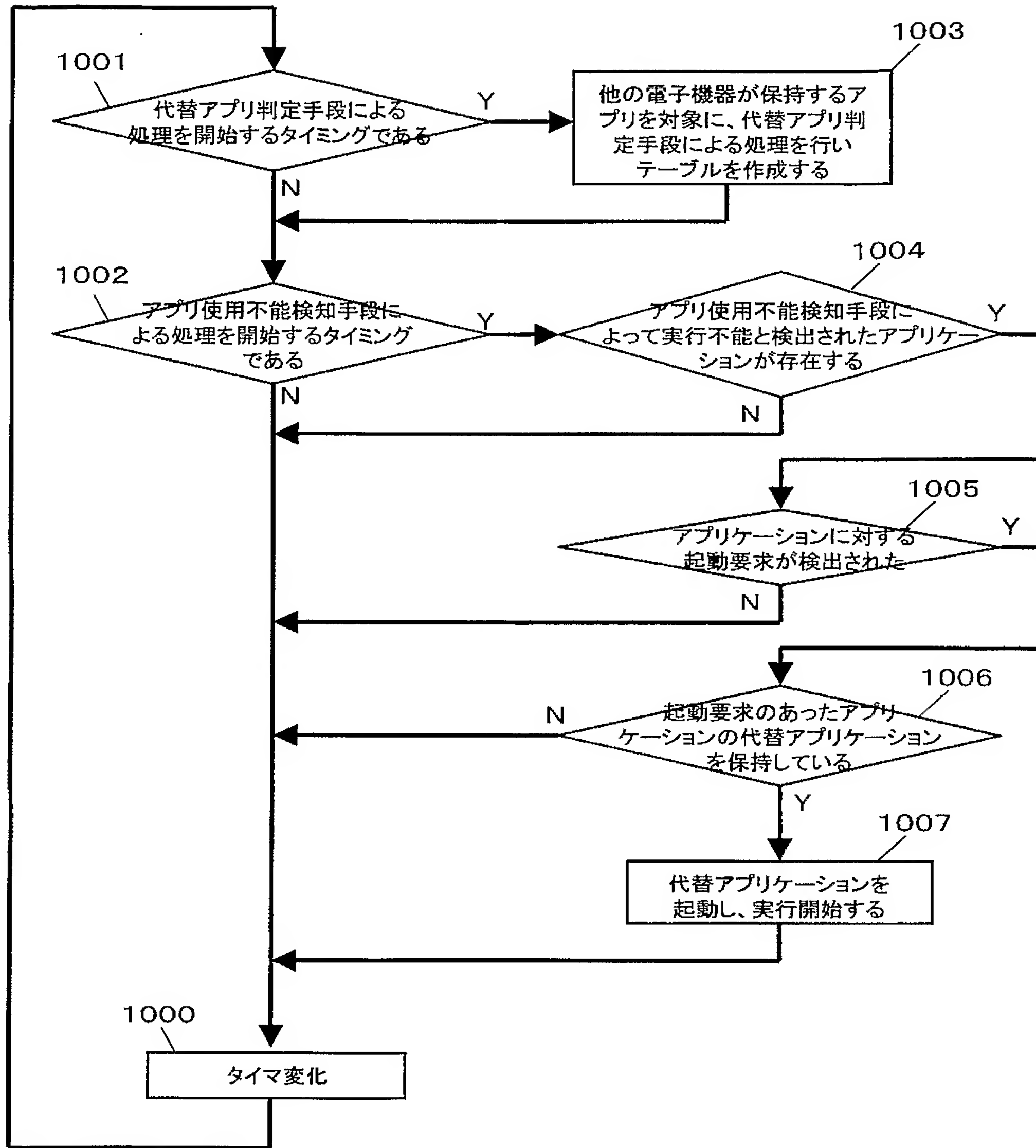
【図 8】



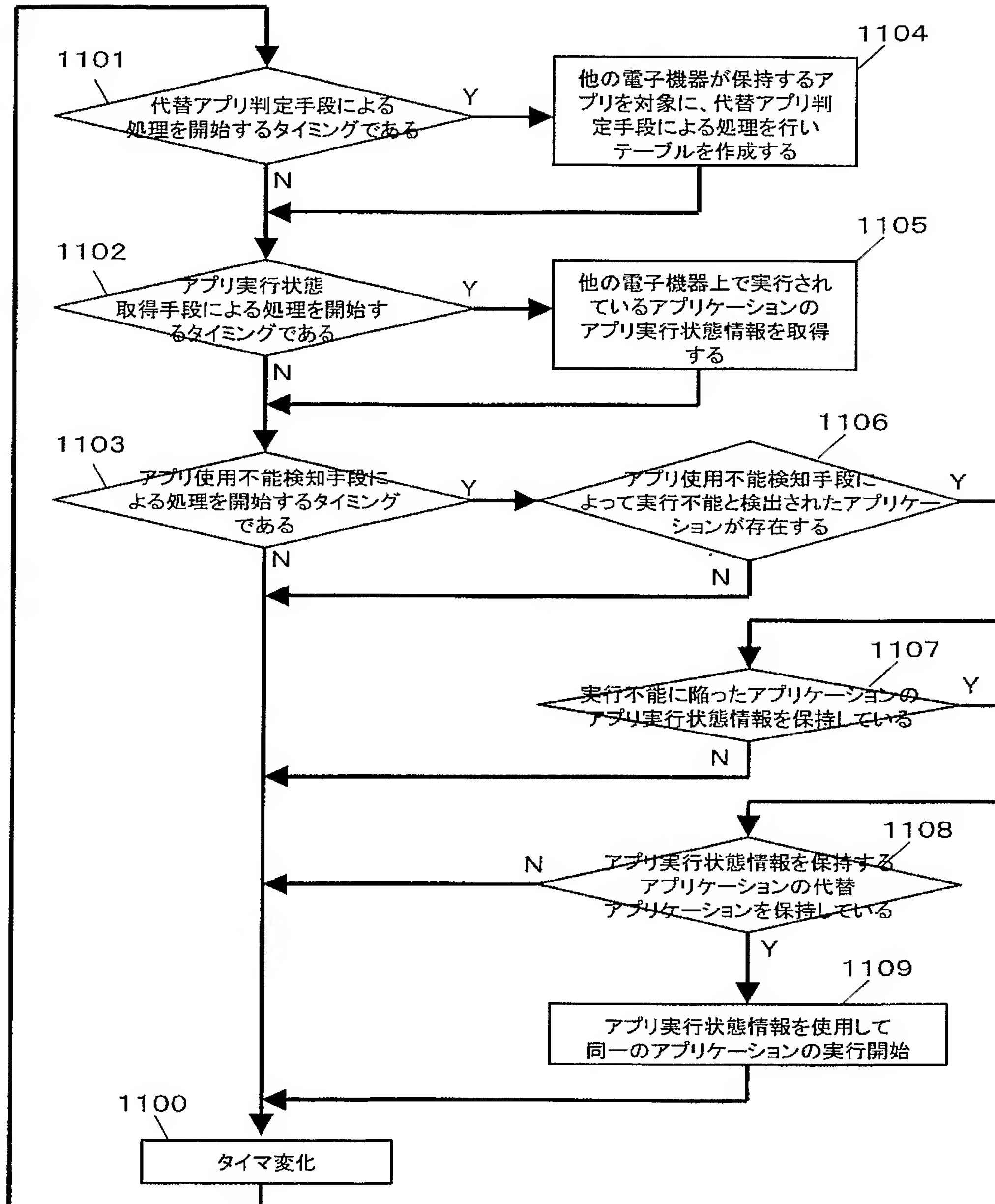
【図 9】



【図 10】



【図 11】



【書類名】 要約書**【要約】**

【課題】 センターサーバによる集中管理方式を前提としない家電ネットワークに対して有効な手段を与えること。またサービスの実施中にそのサービスの継続が不可能な状態に陥った場合の復旧に関して有効な手段を与えること。

【解決手段】 機器にアプリケーションの同一性判定手段、代替アプリ判定手段を設けることで、センターサーバを前提としない、柔軟性の高いサービスの復旧手段を提供する。またアプリ実行状態取得手段により、サービスの実行状態の情報を機器間で共有することで、中断されたサービスの継続を可能にする。家電ネットワークに接続された機器が保持するサービスを用いて、一つの機器上で発生したサービスの実行不能状態を検知し、別の機器によってサービスを代替あるいは継続するための装置を提供する。

【選択図】 図 1

特願 2 0 0 4 - 0 3 1 7 6 1

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 5 8 2 1]

1. 変更年月日

1 9 9 0 年 8 月 2 8 日

[変更理由]

新規登録

住 所

大阪府門真市大字門真 1 0 0 6 番地

氏 名

松下電器産業株式会社